

10/18/99
JC520 U.S. PTO

Please type a plus sign (+) inside this box [+]

PTO/SB/05 (12/97)

Approved for use through 09/30/00. OMB 0651-0032

Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No. 80398.P282

Total Pages 2

First Named Inventor or Application Identifier Franz, et al.

Express Mail Label No. EL234218299US

JC503 U.S. PTO
09/420509
10/16/99

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, D. C. 20231

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

1. Fee Transmittal Form
(Submit an original, and a duplicate for fee processing)
2. Specification (Total Pages 40)
(preferred arrangement set forth below)
 - Descriptive Title of the Invention
 - Cross References to Related Applications
 - Statement Regarding Fed sponsored R & D
 - Reference to Microfiche Appendix
 - Background of the Invention
 - Brief Summary of the Invention
 - Brief Description of the Drawings (if filed)
 - Detailed Description
 - Claims
 - Abstract of the Disclosure
3. Drawings(s) (35 USC 113) (Total Sheets 7)
4. Oath or Declaration (Total Pages 5)
 - a. Newly Executed (Original or Copy)
 - b. Copy from a Prior Application (37 CFR 1.63(d))
(for Continuation/Divisional with Box 17 completed) (Note Box 5 below)
 - i. DELETIONS OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
5. Incorporation By Reference (useable if Box 4b is checked)
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
6. Microfiche Computer Program (Appendix)

7. Nucleotide and/or Amino Acid Sequence Submission
(if applicable, all necessary)
a. Computer Readable Copy
b. Paper Copy (identical to computer copy)
c. Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. Assignment Papers (cover sheet & documents(s))
9. a. 37 CFR 3.73(b) Statement (where there is an assignee)
 b. Power of Attorney
10. English Translation Document (if applicable)
11. a. Information Disclosure Statement (IDS)/PTO-1449
 b. Copies of IDS Citations
12. Preliminary Amendment
13. Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
14. a. Small Entity Statement(s)
 b. Statement filed in prior application, Status still proper and desired
15. Certified Copy of Priority Document(s) (if foreign priority is claimed)
16. Other: _____

17. If a **CONTINUING APPLICATION**, check appropriate box and supply the requisite information:
 Continuation Divisional Continuation-in-part (CIP)
of prior application No: _____

18. **Correspondence Address**

Customer Number or Bar Code Label _____ (Insert Customer No. or Attach Bar Code Label here)
or
 Correspondence Address Below

NAME Ronald C. Card, Reg. 44,687  10/18/89
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

ADDRESS 12400 Wilshire Boulevard
Seventh Floor

CITY Los Angeles STATE California ZIP CODE 90025-1026

Country U.S.A. TELEPHONE (408) 720-8598 FAX (408) 720-9397

UNITED STATES PATENT APPLICATION
FOR
**SYSTEM AND METHOD TO MATCH LINGUISTIC STRUCTURES USING
THESAURUS INFORMATION**

INVENTORS:

ALEXANDER FRANZ

KEIKO HORIZUCHI

KAZAMI UCHIDA

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026

(408) 720-8598

"Express Mail" mailing label number EL234218299US

Date of Deposit: 10/18/99

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

Cheri Clinkenbeard

(Typed or printed name of person mailing paper or fee)

Clinkenbeard

(Signature of person mailing paper or fee)

10/18/99

Date

SYSTEM AND METHOD TO MATCH LINGUISTIC STRUCTURES USING THESAURUS INFORMATION

BACKGROUND OF THE INVENTION

5 With the continuing growth of multinational business dealings where the global economy brings together business people of all nationalities and with the ease and frequency of today's travel between countries, the demand for a machine-aided interpersonal communication system that provides accurate near real-time language translation is a compelling need.

10 This system would relieve users of the need to possess specialized linguistic or translational knowledge.

15 A typical language translation system functions by using natural language processing. Natural language processing is generally concerned with the attempt to recognize a large pattern or sentence by decomposing it into small subpatterns according to linguistic rules. A natural language processing system uses considerable knowledge about the structure of the language, including what the words are, how words combine to form sentences, what the words mean, how word meanings are related to each other, and how word meanings contribute to sentence meanings. Specifically, phonetic and

20 phonological knowledge concerns how words are related to sounds that realize them. Morphological knowledge concerns how words are constructed from more basic units called morphemes. Syntactic knowledge concerns how words can be put together to form correct sentences and determines what structural role each word plays in the sentence and what phrases are subparts of what other phrases. Typical syntactic representations of language are based on the notion of context-free grammars, which represent sentence structure in terms of what phrases are subparts of other phrases. This syntactic information is often presented in a tree form. Semantic knowledge concerns what words mean and how these meanings combine in sentences to form sentence meanings. This is

the study of context-independent meaning – the meaning a sentence has regardless of the context in which it is used.

Natural language processing systems further comprise interpretation processes that map from one representation to the other. For instance, the process that maps a sentence to its syntactic structure is called parsing, and it is performed by a component called a parser. The parser uses knowledge about word and word meaning, the lexicon, and a set of rules defining the legal structures, the grammar, in order to assign a syntactic structure to an input sentence.

10 Formally, a context-free grammar of a language is a four-tuple comprising nonterminal vocabularies, terminal vocabularies, a finite set of production rules, and a starting symbol for all productions. The nonterminal and terminal vocabularies are disjoint. The set of terminal symbols is called the vocabulary of the language. Pragmatic knowledge concerns how sentences are
15 used in different situations and how use affects the interpretation of the sentence.

Identified problems with previous approaches to natural language processing are numerous. One previous approach uses a thesaurus to calculate semantic distances between linguistic structures. The thesaurus has a fixed
20 structure including four layers, wherein each layer is assumed to represent a fixed unit with a numeric value of 1/3. However, semantic links encoded in a thesaurus do not always represent the same semantic distance. As a result, this previous approach will sometimes yield erroneous values for semantic similarity. Furthermore, a thesaurus with such a fixed structure may not be
25 readily available for all languages.

Another previous approach uses a negative log likelihood of the most informative thesaurus concept – $\log p(c)$ – to calculate semantic similarity between two words.

Therefore, what is required is a thesaurus that may be used with any language, making natural language processing more efficient and accurate. Also, what is required is a method using such a thesaurus for evaluating similarity among words and phrases, and their linguistic representations.

SUMMARY OF THE INVENTION

A system and method for evaluating similarity among multiple data structures are provided. In one embodiment, a number of entries are created in a database and a relative entropy value corresponding to each entry is calculated. Each data structure is analyzed and at least one substructure is generated for each data structure as a result of the analysis. This at least one substructure pertaining to each data structure is matched to the entries in the database to obtain matching entries. Finally, a match value is generated using the relative entropy values corresponding to the matching entries.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of one embodiment for an architecture of a computer system.

Figure 2a is a block diagram of one embodiment for a natural language 5 translation system.

Figures 2b, 2c, and 2d are exemplary diagrams of structures used by the natural language translation system of Figure 2a.

Figures 3a and 3b are exemplary diagrams of portions of a thesaurus hierarchy.

10 Figure 4 is a flow diagram of one embodiment for creating a thesaurus structure.

Figure 5 is a flow diagram of one embodiment for evaluating similarity among structures.

15 Figure 6 is a block diagram of one embodiment of a thesaurus matching system.

DETAILED DESCRIPTION

A system and method for evaluating similarity among multiple data structures are described. In one embodiment, a system and method for evaluating similarity among a number of data structures, such as feature structures, are provided. Although the present invention relies on feature structures, it will be evident to one skilled in the art that any data structure may be used with the present invention.

In the following detailed description of embodiments of the invention, reference is made to the accompanying drawings in which like references 10 indicate similar elements, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. Numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that the present invention may be practiced without these specific details. In some instances, 15 well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring the present invention. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical, electrical and other changes may 20 be made without departing from the scope of the present invention.

Some portions of the detailed descriptions that follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts 25 to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of acts leading to a desired result. The acts are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being 30 stored, transferred, combined, compared, and otherwise manipulated. It has

proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms
5 are to be associated with the appropriate physical quantities and are merely
convenient labels applied to these quantities. Unless specifically stated
otherwise as apparent from the following discussion, it is appreciated that
throughout the description, discussions utilizing terms such as "processing" or
"computing" or "calculating" or "determining" or "displaying" or the like, refer
10 to the action and processes of a computer system, or similar electronic
computing device, that manipulates and transforms data represented as
physical (electronic) quantities within the computer system's registers and
memories into other data similarly represented as physical quantities within the
computer system memories or registers or other such information storage,
15 transmission or display devices.

The present invention also relates to apparatus for performing the
operations herein. This apparatus may be specially constructed for the required
purposes, or it may comprise a general purpose computer selectively activated
or reconfigured by a computer program stored in the computer. Such a
20 computer program may be stored in a computer readable storage medium, such
as, but is not limited to, any type of disk including floppy disks, optical disks,
CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random
access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or
any type of media suitable for storing electronic instructions, and each coupled
25 to a computer system bus.

The algorithms and displays presented herein are not inherently related
to any particular computer or other apparatus. Various general purpose
systems may be used with programs in accordance with the teachings herein, or
it may prove convenient to construct more specialized apparatus to perform the
30 required method. For example, any of the methods according to the present

invention can be implemented in hard-wired circuitry, by programming a general purpose processor or by any combination of hardware and software.

One of skill in the art will immediately appreciate that the invention can be practiced with computer system configurations other than those described

5 below, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention can also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. The
10 required structure for a variety of these systems will appear from the description below.

The methods of the invention are described in terms of computer software. If written in a programming language conforming to a recognized standard, sequences of instructions designed to implement the methods can be
15 compiled for execution on a variety of hardware platforms and for interface to a variety of operating systems. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein. Furthermore, it is common in
20 the art to speak of software, in one form or another (e.g., program, procedure, application...), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a computer causes the processor of the computer to perform an action or produce a result.

Figure 1 is a block diagram of one embodiment for an architecture of a
25 computer system 100. Referring to Figure 1, computer system 100 includes system bus 101 that allows for communication among processor 102, digital signal processor 108, memory 104, and non-volatile storage device 107. System bus 101 may also receive inputs from keyboard 122, pointing device 123, and speech signal input device 125. System bus 101 provides outputs to display
30 device 121, hard copy device 124, and output device 126 (such as, for example,

an audio speaker). Memory 104 may include, for example, read only memory (ROM), random access memory (RAM), flash memory, or any combination of the above.

It will be appreciated that computer system 100 may be controlled by 5 operating system software which includes a file management system, such as, for example, a disk operating system, which is part of the operating system software. The file management system may be stored in non-volatile storage device 107 and may be configured to cause processor 102 to execute the various functions required by the operating system to input and output data and to 10 store data in memory 104 and on non-volatile storage device 107.

Figure 2a is a block diagram of one embodiment for a natural language 15 translation system 200. Referring to Figure 2a, natural language translation system 200 includes five modules, supporting databases, and associated grammars to quickly and accurately translate text between source and target 20 languages. Input text may be directly input into natural language translation system 200 (for example, as with a person typing sentences into a computer using keyboard 122). Alternatively, input text to natural language translation system 200 may be the output of another system, such as, for example, output from a speech recognition system (for example, speech input device 125), or from an optical character recognition system (not shown).

An English sentence "He wants to go to the White House" is used 25 throughout this section as example text input to describe the functioning of the system 200. The individual units in a sentence are referred to herein as "words" but the natural language translation system 200 is not limited to only word-based natural languages, having equal applicability to translation of character-based languages as well. Except where the differences in processing word-based and character-based languages are specified, the term "word" is intended to encompass both words and characters.

In the following description, a grammar is generally a set of context-free 30 rules that define the valid phrase structures in the source or target language,

with each context-free rule associated with one or more statements (the “rule body”) that perform tests and manipulations on the linguistic representations (feature structures). Thus, an English sentence may be combined from a noun phase (NP) and a verb phrase (VP), but the subject and verb forms must agree,

5 e.g., “He want to go to the White House” is a valid phrase structure but an improper English sentence. All rule bodies utilized by the grammars of language translation system 200 are in the form of computer-executable routines produced by defining the grammar in terms of a grammar programming language (GPL) and passing appropriate rule bodies (209, 215, 10 219, and 225) through a GPL compiler 240. The output of the GPL compiler 240 may be in the form of directly executable code, or may be in the form of standard computer programming language statements (such as, for example, C, C++, Pascal, or Lisp) which are then input into the corresponding programming language compiler to produce executable code. In either case, the compiled 15 grammars include a specific function for each context-free rule. The specific function performs all the processing required by the rule and its associated rule body. Furthermore, the interfaces between the compiled grammars and the modules enable a single language translation system 200 to perform translation between multiple natural languages, and to perform more than one translation 20 simultaneously.

A morphological analysis module 206 takes text input 202 and uses a source language dictionary 204 to decompose the words into morphemes by identifying root forms, grammatical categories, thesaurus information, and other lexical features of the words. The morphological analysis module 206 25 builds a “feature structure” for each word. Feature structures are well known in the art as linguistic data structures that contain feature-value pairs for strings, symbols, and numbers that appear in a natural language sentence. Each feature of a word is mapped to the appropriate value through a function commonly designated as:

word \mapsto [feature : value]

Thus, a simplified, exemplary representation of the feature structures for the words "he" and "wants" are as follows:

(Feature Structure 1) $I \mapsto \begin{bmatrix} \text{root : he} \\ \text{cat : pronoun} \end{bmatrix}$

5 (Feature Structure 2) $\text{wants} \mapsto \begin{bmatrix} \begin{bmatrix} \text{root : want} \\ \text{cat : noun} \end{bmatrix} \\ \text{OR} \\ \begin{bmatrix} \text{root : want} \\ \text{cat : verb} \end{bmatrix} \end{bmatrix}$

The Feature Structure 2 may be referred to as a "disjunctive" feature structure as it represents two mutually exclusive feature structures that are valid for the word.

It will be appreciated that the grammatical category is not the only feature of 10 these two words and that morphological analysis module 206 outputs full feature structures. The example feature structures are simplified for the sake of clarity in explanation and are also frequently represented by a shorthand notation, e.g., [want] or NP[].

The feature structures built by morphological analysis module 206 are 15 input into lexical ambiguity reduction module 210. In one embodiment, lexical ambiguity reduction module 210 may segment the words in character-based languages that do not utilize spaces through a database of lexical connector feature rules 208. Lexical connector feature rules 208 are created from GPL grammar statements as described above. Each possible combination of adjacent 20 segmented words are assigned a lexical cost. Dictionary 204 defines combinations of words ("multiwords"). Lexical ambiguity reduction module 210 evaluates each feature structures that contains a part-of-speech (POS) ambiguity, such as the feature structure for the word "wants" which is tagged as both a noun and a verb. The various possible POS tags are assigned a lexical 25 cost. Lexical ambiguity reduction module 210 weighs the cost assigned to each

word in the sentence and selects those feature structures that have the lowest cost.

The feature structures chosen for the words by lexical ambiguity reduction module 210 are passed to syntactic analysis module 216. Syntactic analysis module 216 combines the chosen feature structures into a feature structure that represents the content of the input sentence. In one embodiment, syntactic analysis module 216 uses parsing grammar 212 to create a syntax parse tree for the sentence. Parsing grammar 212 contains the source language context-free grammar rules in the form of a parsing table and the associated rule bodies in executable code. Each leaf of the syntax parse tree is a feature structure for one of the words in the sentence. Once the leaves are created, an intermediate feature structure for each branch (parent) node in the syntax parse tree is built by combining its child nodes as specified in one or more of the context-free grammar rules. The rule body for each potentially applicable context-free grammar rule manipulates the various feature structures at the child nodes and determines whether the associated context-free rule could create a valid phrase from the possible combinations. A rule body may cause a thesaurus 214 to be queried as part of the manipulation. It will be appreciated that the feature structure that results from applying the context-free grammar rules may be nested (i.e., contain multiple feature structures from each child node). Syntactic analysis module 216 may create the syntax parse tree shown in Figure 2b for the example sentence from its constituent feature structures, with the following feature structure at the top (root) of the syntax parse tree to represent the full sentence:

25 (Feature Structure 3) $S \rightarrow \begin{bmatrix} \text{SUBJ "he"} \\ \text{VERB "wants to go"} \\ \text{OBJ "to the White House"} \end{bmatrix}$

It will be appreciated that both the syntax parse tree 250 and the Feature Structure 3 are not exact representations but are simplified for purposes of ease in explanation.

The feature structure for the sentence in the source language is passed to transfer module 222. The feature structure represents the analysis of the source input and may contain a number of nested linguistic representations (referred herein as sub-structures or slots). Transfer module 222 uses transfer grammar

5 218 to match source language slots of the input with source language slots in example database 220. Example database 220 contains feature structure pairs in the source language and a target language. For example, database 220 may contain matching feature structures in English and Japanese. Transfer grammar 218 consists of a set of rewrite rules with a context-free component and a GPL
10 rule body. The context-free parts of the rules are used to create a transfer generation tree.

Transfer module 222 uses the GPL rule bodies within transfer grammar 218 to match the input source sub-structures or slots to the source sub-structures or slots in example database 220. If a good match is found (in one
15 embodiment, a low overall match cost), transfer module 222 checks if all sub-structures or slots of the input feature structure have found a match. If a match for a sub-structure is not found, the sub-structure is used as input to transfer module 222. A transfer generation tree of the form shown in Figure 2c is used to break the sub-structure into multiple sub-structures. The new input may be a
20 part of the original, source feature structure or a new feature sub-structure that is constructed from sections of different slots.

Transfer module 222 uses the input feature structure (or sub-structure) in the source language as the starting symbol to build transfer generation tree 260. Root 261 is a symbol-node (s-node) and is labeled with the starting symbol of

25 the feature structure. The transfer grammar determines which transfer grammar rules are applicable to the feature structure at the root 261, and creates child rule-node(s) (r-node) 263 depending from root 261. In one embodiment, r-nodes 263 are the rule numbers within transfer grammar 218 that may be validly applied to the input feature structure. Transfer grammar 218 rules
30 added to tree 260 are applied to the s-nodes 265. If the application of each rule

succeeds, a child rule-node (r-node) 265 is added to tree 260. If the application fails, the s-node 261 is tagged as "dead" for sub-sequent removal. Transfer grammar 218 then creates a new s-node 265 for each r-node 263. Again, the applicable rules are found for each s-node 265 and applied. The process is

5 repeated until all sub-features within the feature structure have been expanded.

Transfer generation tree 260 is then pruned to remove any "dead" nodes and corresponding sub-trees. If root 261 is tagged as "dead," the generation fails.

Otherwise, the resulting transfer generation tree 260 is used by transfer module 222 to match the feature structure against the example database 220. The

10 feature structures and sub-structures in the target language associated with a match are substituted for the corresponding feature structures and sub-

structures matched in the source language. Transfer module 222 recursively applies the transfer rules contained within transfer grammar 218 from the top-most transfer rules until all meaningful sub-features or constituents within the

15 input source feature structure are transferred to the target sub-structures. The transfer module 222 will consult the thesaurus 214 when required to do so by a transfer rule. Transfer module 222 outputs a feature structure in the target language.

The feature structure for the sentence in the target language is passed to

20 a morphological and syntactical generation module 228, where it is used as the root node for a syntactical generation tree, an example of which is shown in Figure 2d. The syntactical generation tree is built in the same fashion as the transfer generation tree, with context-free rules in a generation grammar 224 as its r-nodes 273. The generation grammar 224 copies information to each s-node

25 275, 279. Unlike the transfer module 226, in which multiple sub-transfers created multiple transfer generation trees, only one syntactical generation tree is created by the morphological and syntactical generation module 228. Any s-node that is not a leaf node 279, i.e., associated with a feature structure for a word, is used to generate the next level of r-nodes. When all child s-nodes

30 under an r-node are leaf nodes, the current branch of the tree is complete and

the morphological and syntactical generation module 228 traverses back up the tree to find the next s-node that is not a leaf node. The thesaurus 214 is consulted when necessary during the generation of the tree. The transfer generation tree is complete when all the lowest level s-node are leaf nodes.

5 When the syntactical generation tree is complete, the leaf nodes contain output feature structures representing the words in one or more translations of the input sentence. The sequence of output feature structures that represents the best sentence is converted into output text 230 by the morphological and syntactical generation module 228 using the dictionary 226. Alternatively, all
10 output feature structures for all sentences may be converted into the output text 230.

In Figures 3a and 3b, a database, such as thesaurus 214, comprises a structure having multiple entries, such as a top node 300, also known as "root" node, and several nodes 304 through 308 connected directly or indirectly to root node 300. The thesaurus structure encodes semantic "is-a" relationships
15 between concepts residing within the thesaurus nodes. For example, Figure 3b illustrates a portion of the thesaurus structure, wherein:

APPLE is-a FRUIT-NAME; and

APPLE is-a BRAND-NAME.

20 In one embodiment, the thesaurus structure is created manually, using a tool having a "user-friendly" graphic interface, such as data entry equipment. In alternate embodiments, other tools for creating the structure may also be used. The creation and further processing of the thesaurus structure will be described in more detail below.

25 In one embodiment, thesaurus 214 shown in Figure 2a is created and processed within system 200. However, in alternate embodiments, thesaurus 214 may be created and processed off-line, that is prior to the processing of system 200. An operator inputs thesaurus concepts and relationships and organizes the concepts and relationships within the thesaurus structure,
30 starting from root node 300. The thesaurus concepts reside within the nodes of

the thesaurus, while branches between adjacent nodes represent the semantic "is-a" relationships between concepts.

As new thesaurus concepts become available and are input within the thesaurus structure, the structure may change due to changes in the relationships between concepts. After the concepts are entered, the thesaurus structure is exported for further processing using a thesaurus link file. The thesaurus link file is created by implementing the concepts residing within the nodes into a format, which contains a series of "is-a" statements, replacing the nodes and branches of the thesaurus structure. An example of a thesaurus link file may be illustrated as follows:

ABSTR is-a ROOT
ABSTR-PARTIT is-a QUANT-PARTIT
ACCESSORY is-a CLOTHING
ACCON is-a STRUCT

15

20
WINTER-EQUIP is-a SPORTS-EQUIP
WORK is-a FEATURE
WRAP is-a MOVE-ARTIF
WRITE-UTENS is-a UTENS

In one embodiment, the creation of the database, such as thesaurus 214, includes creation of the multiple entries in the database, such as the nodes, and calculation of a relative entropy value corresponding to each entry. Figure 4 is a flow diagram of one embodiment for creating the thesaurus structure and exporting the thesaurus link file. Referring to Figure 4, initially at processing block 410, the thesaurus structure is created. At processing blocks 420 and 430, the thesaurus link file is read and a validity check of the semantic "is-a" relationships is performed. The validity check shown at processing block 430

includes several operations, such as, for example, a check for cycles and a check that every thesaurus concept has in fact a parent concept within the thesaurus structure. In alternate embodiments, other operations may be included in the validity check of the thesaurus link file.

5 At processing block 440, the lexical probability files are read, if available. The lexical probability files contain information about the probability of each word occurring in the domain of translation. If the lexical probability files are not available for reading and compiling, all words are assumed to occur with the same probability in the domain of translation.

10 At processing block 450, a lexicon database having multiple lexicon entries, containing elements, such as words, is read and possible thesaurus concepts associated with each lexicon entry are assigned. For example, the English noun lexicon entry for "apple" shown below:

((CAT NOUN)
15 (ROOT "apple")
(LEX-TYPE EXT)
(THES FRUIT-NAME BRAND-NAME APPLY)
(V-INIT+)
(MAYFLOWER-INPUT+))

20 contains a thesaurus code THES FRUIT-NAME BRAND-NAME indicating possible semantic concepts for the word "apple". Similarly, lexicon entries in languages other than English contain thesaurus codes indicating respective semantic concepts for the words.

At processing block 460, scaled relative entropy values are calculated for 25 each node of the thesaurus structure. Each node corresponding to a thesaurus concept has its own entropy value, Entropy (Node). The scaled relative entropy value corresponding to a node within the thesaurus structure is calculated using the formula:

$$\text{RelativeEntropy}(\text{Node}) = k * \frac{\text{Entropy}(\text{Node})}{\text{Entropy}(\text{RootNode})}$$

wherein Entropy (Root Node) is the entropy value corresponding to the root node and k is a constant factor. The entropy of a node, Entropy (Node), is

5 further calculated based on all the words dominated by that node with the formula:

$$\text{Entropy}(\text{Node}) = - \sum_{\text{words dominated by Node}} p(\text{word}) \log p(\text{word})$$

At processing block 470, after the scaled relative entropy value corresponding to each node in the thesaurus structure is calculated, that value

10 is stored, together with the corresponding node, within the thesaurus structure. In one embodiment, the values and nodes are stored in a compressed format. The thesaurus 214 and its structure may now be used within system 200 for further evaluation of the feature structures.

Figure 5 is a flow diagram of one embodiment for evaluating similarity

15 between feature structures. At processing block 510, feature structures are received. In addition, the feature structures are analyzed into multiple substructures. At processing block 520, thesaurus codes THES (...) are retrieved from the substructures (in a prior procedure, the feature structures are compared to the lexicon entries within the lexicon database, and the thesaurus 20 codes corresponding to the lexicon entries matching the feature structures are assigned to the feature structures and their substructures).

At processing block 530, the thesaurus codes are used to search the thesaurus structure for nodes matching each thesaurus code. In one embodiment, the search returns the most specific thesaurus node corresponding 25 to each substructure. In one embodiment, the most specific thesaurus node is the lowest common parent that dominates the respective thesaurus code within the thesaurus concept structure.

At processing block 540, the relative entropy value corresponding to the most specific thesaurus node returned for each substructure is retrieved and sent to the syntactic analysis module 216. Finally, at processing block 550, a match cost value is calculated using the retrieved relative entropy values. In 5 one embodiment, the match cost is calculated as a sum of the relative entropy values of each returned node and is returned to the system 200 at processing block 560. In alternate embodiments, other formulas may be used to calculate the match cost.

A system for evaluating similarity among multiple data structures is 10 further described in more detail. In Figure 6, a thesaurus matching system 600 comprises analyzer 610, matching unit 620, entropy calculator 630, lexicon database 640, such as a dictionary, and database with multiple entries 650, such as a thesaurus structure.

Analyzer 610 receives feature structures and analyzes the structures into 15 substructures for further processing. Matching unit 620 retrieves thesaurus codes from each substructure. Matching unit 620 also matches the retrieved thesaurus codes with nodes within the thesaurus structure, retrieving the most specific node corresponding to each substructure. Matching unit 620 retrieves an associated relative entropy value from thesaurus 650 corresponding to the 20 most specific node of each substructure, sending the relative entropy values into entropy calculator 630. Entropy calculator 630 calculates a match cost value using the retrieved relative entropy values.

Thus, a system and method for evaluating similarity among a plurality 25 of data structures has been provided. Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention as set forth in the claim. Accordingly, the specification and drawings are to be recorded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

- 1 1. A method for evaluating similarity among a plurality of data
2 structures, comprising:
 - 3 analyzing each structure of said plurality of data structures to generate
4 at least one substructure;
 - 5 matching said at least one substructure to a database having a plurality
6 of entries to obtain at least one matching entry; and
 - 7 generating a match value using a relative entropy value corresponding
8 to said at least one matching entry.
- 1 2. The method according to claim 1, further comprising:
 - 2 creating said plurality of entries in said database; and
 - 3 processing said plurality of entries in said database.
- 1 3. The method according to claim 2, wherein said creating further
2 comprises creating said plurality of entries using a tool having a graphical user
3 interface and exporting said plurality of entries to said database.
- 1 4. The method according to claim 2, wherein said processing further
2 comprises:
 - 3 verifying said plurality of entries for validity; and
 - 4 calculating said relative entropy value corresponding to each entry of
5 said plurality of entries.

1 5. The method according to claim 4, wherein said processing further
2 comprises storing said each entry of said plurality of entries together with said
3 corresponding relative entropy value in a compressed format.

1 6. The method according to claim 1, further comprising extracting
2 from a lexicon database having a plurality of elements each element associated
3 to said each structure, assigning at least one code of said each element to said
4 each structure, and retrieving said at least one code during matching to obtain
5 said at least one matching entry.

1 7. The method according to claim 6, further comprising reading
2 lexical probability files and assigning a probability value to said each element of
3 said plurality of elements in said lexicon database.

1 8. The method according to claim 1, wherein each structure of said
2 plurality of data structures is a representation of a linguistic expression.

1 9. The method according to claim 4, wherein said database is a
2 thesaurus hierarchy including a root entry, said plurality of entries depending
3 from said root entry.

1 10. The method according to claim 9, wherein said relative entropy
2 value corresponding to said each entry of said plurality of entries is calculated
3 based on an entropy value of said each entry and an entropy value of said root
4 entry.

1 11. The method according to claim 6, wherein said each element in
2 said lexicon database is a word.

1 12. A method for evaluating similarity among a plurality of data
2 structures comprising:
3 creating a plurality of entries in a database; and
4 calculating a relative entropy value corresponding to each entry of said
5 plurality of entries.

1 13. The method according to claim 12, further comprising storing said
2 each entry of said plurality of entries together with said corresponding relative
3 entropy value in a compressed format.

1 14. The method according to claim 12, further comprising:
2 creating said plurality of entries using a tool having a graphical user
3 interface; and
4 exporting said plurality of entries to said database.

1 15. The method according to claim 12 further comprising:
2 analyzing each structure of said plurality of data structures to generate
3 at least one substructure;
4 matching said at least one substructure of said each structure to said
5 database to obtain at least one matching entry; and
6 generating a match value using said relative entropy value
7 corresponding to said at least one matching entry.

1 16. The method according to claim 15, further comprising:

2 verifying said plurality of entries for validity;
3 extracting from a lexicon database having a plurality of elements each
4 element associated to said each structure;
5 reading lexical probability files;
6 assigning a probability value to said each element of said plurality of
7 elements in said lexicon database;
8 assigning at least one code of said each element to said each structure;
9 and
10 retrieving and matching said at least one code to said database to obtain
11 said at least one matching entry.

1 17. The method according to claim 16, wherein said each structure of
2 said plurality of data structures is a representation of a linguistic expression.

1 18. The method according to claim 12, wherein said database is a
2 thesaurus hierarchy including a root entry, said plurality of entries depending
3 from said root entry.

1 19. The method according to claim 18, wherein said relative entropy
2 value corresponding to said each entry of said plurality of entries is calculated
3 based on an entropy value for said each entry and an entropy value for said
4 root entry.

1 20. The method according to claim 15, wherein said each element in
2 said lexicon database is a word.

1 21. A computer readable medium containing executable instructions
2 which, when executed in a processing system, cause the system to perform a
3 method for evaluating similarity among a plurality of data structures, the
4 method comprising:

5 analyzing each structure of said plurality of data structures to generate
6 at least one substructure;

7 matching said at least one substructure to a database having a plurality
8 of entries to obtain at least one matching entry; and

9 generating a match value using a relative entropy value corresponding
10 to said at least one matching entry.

1 22. The computer readable medium according to claim 21, wherein
2 the method further comprises:

3 creating said plurality of entries in said database; and
4 processing said plurality of entries in said database.

1 23. The computer readable medium according to claim 22, wherein
2 said creating further comprises creating said plurality of entries using a tool
3 having a graphical user interface and exporting said plurality of entries to said
4 database.

1 24. The computer readable medium according to claim 22, wherein
2 said processing further comprises:

3 verifying said plurality of entries for validity; and
4 calculating said relative entropy value corresponding to each entry of
5 said plurality of entries.

1 25. The computer readable medium according to claim 24, wherein
2 said processing further comprises storing said each entry of said plurality of
3 entries together with said corresponding relative entropy value in a
4 compressed format.

1 26. The computer readable medium according to claim 21, further
2 comprising extracting from a lexicon database having a plurality of elements
3 each element associated to said each structure, assigning at least one code of
4 said each element to said each structure, and retrieving said at least one code
5 during matching to obtain said at least one matching entry.

1 27. The computer readable medium according to claim 26, further
2 comprising reading lexical probability files and assigning a probability value to
3 said each element of said plurality of elements in said lexicon database.

1 28. The computer readable medium according to claim 21, wherein
2 each structure of said plurality of data structures is a representation of a
3 linguistic expression.

1 29. The computer readable medium according to claim 24, wherein
2 said database is a thesaurus hierarchy including a root entry, said plurality of
3 entries depending from said root entry.

1 30. The computer readable medium according to claim 29, wherein
2 said relative entropy value corresponding to said each entry of said plurality of
3 entries is calculated based on an entropy value of said each entry and an
4 entropy value of said root entry.

1 31. The computer readable medium according to claim 26, wherein
2 said each element in said lexicon database is a word.

1 32. A computer readable medium containing executable instructions
2 which, when executed in a processing system, cause the system to perform a
3 method for evaluating similarity among a plurality of data structures, the
4 method comprising:

5 creating a plurality of entries in a database; and
6 calculating a relative entropy value corresponding to each entry of said
7 plurality of entries.

1 33. The computer readable medium according to claim 32, further
2 comprising storing said each entry of said plurality of entries together with said
3 corresponding relative entropy value in a compressed format.

1 34. The computer readable medium according to claim 32, further
2 comprising:
3 creating said plurality of entries using a tool having a graphical user
4 interface; and
5 exporting said plurality of entries to said database.

1 35. The computer readable medium according to claim 32 further
2 comprising:
3 analyzing each structure of said plurality of data structures to generate
4 at least one substructure;
5 matching said at least one substructure of said each structure to said
6 database to obtain at least one matching entry; and

7 generating a match value using said relative entropy value
8 corresponding to said at least one matching entry.

1 36. The computer readable medium according to claim 35, further
2 comprising:

3 verifying said plurality of entries for validity;
4 extracting from a lexicon database having a plurality of elements each
5 element associated to said each structure;
6 reading lexical probability files;
7 assigning a probability value to said each element of said plurality of
8 elements in said lexicon database;
9 assigning at least one code of said each element to said each structure;
10 and
11 retrieving and matching said at least one code to said database to obtain
12 said at least one matching entry.

1 37. The computer readable medium according to claim 36, wherein
2 said each structure of said plurality of data structures is a representation of a
3 linguistic expression.

1 38. The computer readable medium according to claim 32, wherein
2 said database is a thesaurus hierarchy including a root entry, said plurality of
3 entries depending from said root entry.

1 39. The computer readable medium according to claim 38, wherein
2 said relative entropy value corresponding to each entry of said plurality of

3 entries is calculated based on an entropy value for said each entry and an
4 entropy value for said root entry.

1 40. The computer readable medium according to claim 35, wherein
2 said each element in said lexicon database is a word.

1 41. An article of manufacture comprising a program storage medium
2 readable by a computer and tangibly embodying at least one program of
3 instructions executable by said computer to perform method steps for
4 evaluating similarity among a plurality of data structures, said method
5 comprising:

6 analyzing each structure of said plurality of data structures to generate
7 at least one substructure;

8 matching said at least one substructure to a database having a plurality
9 of entries to obtain at least one matching entry; and

10 generating a match value using a relative entropy value corresponding
11 to said at least one matching entry.

1 42. The article of manufacture according to claim 41, wherein the
2 method further comprises:

3 creating said plurality of entries in said database; and
4 processing said plurality of entries in said database.

1 43. The article of manufacture according to claim 42, wherein said
2 creating further comprises creating said plurality of entries using a tool having
3 a graphical user interface and exporting said plurality of entries to said
4 database.

1 44. The article of manufacture according to claim 42, wherein said
2 processing further comprises:

3 verifying said plurality of entries for validity; and

4 calculating said relative entropy value corresponding to each entry of
5 said plurality of entries.

1 45. The article of manufacture according to claim 44, wherein said
2 processing further comprises storing each entry of said plurality of entries
3 together with said corresponding relative entropy value in a compressed
4 format.

1 46. The article of manufacture according to claim 41, wherein the
2 method further comprises:

3 extracting from a lexicon database having a plurality of elements each
4 element associated to said each structure;

5 assigning at least one code of said each element to said each structure;
6 and

7 retrieving said at least one during matching to obtain said at least one
8 matching entry.

1 47. The article of manufacture according to claim 46, wherein the
2 method further comprises reading lexical probability files and assigning a
3 probability value to said each element of said plurality of elements in said
4 lexicon database.

1 48. The article of manufacture according to claim 41, wherein each
2 structure of said plurality of data structures is a representation of a linguistic
3 expression.

1 49. The article of manufacture according to claim 44, wherein said
2 database is a thesaurus hierarchy including a root entry, said plurality of entries
3 depending from said root entry.

1 50. The article of manufacture according to claim 49, wherein said
2 relative entropy value corresponding to said each entry of said plurality of
3 entries is calculated based on an entropy value of said each entry and an
4 entropy value of said root entry.

1 51. The article of manufacture according to claim 46, wherein said
2 each element in said lexicon database is a word.

1 52. An article of manufacture comprising a program storage medium
2 readable by a computer and tangibly embodying at least one program of
3 instructions executable by said computer to perform method steps for
4 evaluating similarity among a plurality of data structures, said method
5 comprising:
6 creating a plurality of entries in a database; and
7 calculating a relative entropy value corresponding to each entry of said
8 plurality of entries.

1 53. The article of manufacture according to claim 52, wherein the
2 method further comprises storing said each entry of said plurality of entries

3 together with said corresponding relative entropy value in a compressed
4 format.

1 54. The article of manufacture according to claim 52, wherein the
2 method further comprises:

3 creating said plurality of entries using a tool having a graphical user
4 interface; and

5 exporting said plurality of entries to said database.

1 55. The article of manufacture according to claim 52, wherein the
2 method further comprises:

3 analyzing each structure of said plurality of data structures to generate
4 at least one substructure;

5 matching said at least one substructure of said each structure to said
6 database to obtain at least one matching entry; and

7 generating a match value using said relative entropy value
8 corresponding to said at least one matching entry.

1 56. The article of manufacture according to claim 55, wherein the
2 method further comprises:

3 verifying said plurality of entries for validity;

4 extracting from a lexicon database having a plurality of elements each
5 element associated to said each structure;

6 reading lexical probability files;

7 assigning a probability value to said each element of said plurality of
8 elements in said lexicon database;

9 assigning at least one code of said each element to said each structure;
10 and
11 retrieving and matching said at least one code to said database to obtain
12 said at least one matching entry.

1 57. The article of manufacture according to claim 56, wherein said
2 structure of said plurality of data structures is a representation of an linguistic
3 expression.

1 58. The article of manufacture according to claim 52, wherein said
2 database is a thesaurus hierarchy including a root entry, said plurality of entries
3 depending from said root entry.

1 59. The article of manufacture according to a claim 58, wherein said
2 relative entropy value corresponding to said each entry of said plurality of
3 entries is calculated based on an entropy value for said each entry and an
4 entropy value for said root entry.

1 60. The article of manufacture according to claim 55, wherein said
2 each element in said lexicon database is a word.

1 61. A system for evaluating similarity among a plurality of data
2 structures, comprising:
3 means for analyzing each structure of said plurality of data structures to
4 generate at least one substructure;
5 means for matching said at least one substructure to a database having a
6 plurality of entries to obtain at least one matching entry; and

7 means for generating a match value using a relative entropy value
8 corresponding to said at least one matching entry.

1 62. The system according to claim 61, further comprising:
2 means for creating said plurality of entries in said database; and
3 means for processing said plurality of entries in said database.

1 63. The system according to claim 62, wherein said creating means
2 further comprises means for creating said plurality of entries using a tool
3 having a graphical user interface and exporting said plurality of entries to said
4 database.

1 64. The system according to claim 62, wherein said processing means
2 further comprises:
3 means for verifying said plurality of entries for validity; and
4 means for calculating said relative entropy value corresponding to each
5 entry of said plurality of entries.

1 65. The system according to claim 64, wherein said processing means
2 further comprises means for storing said each entry of said plurality of entries
3 together with said corresponding relative entropy value in a compressed
4 format.

1 66. The system according to claim 61, further comprising:
2 means for extracting from a lexicon database having a plurality of
3 elements each element associated to said each structure;

4 means for assigning at least one code of said each element to said each
5 structure; and

6 means for retrieving said at least one code during matching to obtain
7 said at least one matching entry.

1 67. The system according to claim 66, further comprising:
2 means for reading lexical probability files; and
3 means for assigning a probability value to said each element of said
4 plurality of elements in said lexicon database.

1 68. The system according to claim 61, wherein each structure of said
2 plurality of data structures is a representation of a linguistic expression.

1 69. The system according to claim 64, wherein said database is a
2 thesaurus hierarchy including a root entry, said plurality of entries depending
3 from said root entry.

1 70. The system according to claim 69, wherein said relative entropy
2 value corresponding to said each entry of said plurality of entries is calculated
3 based on an entropy value of said each entry and an entropy value of said root
4 entry.

1 71. The system according to claim 66, wherein said each element in
2 said lexicon database is a word.

1 72. A system for evaluating similarity among a plurality of data
2 structures, comprising:

3 means for creating a plurality of entries in a database; and
4 means for calculating a relative entropy value corresponding to each
5 entry of said plurality of entries.

1 73. The system according to claim 72, further comprising means for
2 storing said each entry of said plurality of entries together with said
3 corresponding relative entropy value in a compressed format.

1 74. The system according to claim 72, further comprising:
2 means for creating said plurality of entries using a tool having a
3 graphical user interface; and
4 means for exporting said plurality of entries to said database.

1 75. The system according to claim 72, further comprising:
2 means for analyzing each structure of said plurality of data structures to
3 generate at least one substructure;
4 means for matching said at least one substructure of said each structure
5 to said database to obtain at least one matching entry; and
6 means for generating a match value using said relative entropy value
7 corresponding to said at least one matching entry.

1 76. The system according to claim 75, further comprising:
2 means for verifying said plurality of entries for validity;
3 means for extracting from a lexicon database having a plurality of
4 elements each element associated to said each structure;
5 means for reading lexical probability files;

6 means for assigning a probability value to said each element of said
7 plurality of elements in said lexicon database;
8 means for assigning at least one code of said each element to said each
9 structure; and
10 means for retrieving and matching said at least one code to said database
11 to obtain said at least one matching entry.

1 77. The system according to claim 76, wherein said each structure of
2 said plurality of data structures is a representation of a linguistic expression.

1 78. The system according to claim 72, wherein said database is a
2 thesaurus hierarchy including a root entry, said plurality of entries depending
3 from said root entry.

1 79. The system according to claim 78, wherein said relative entropy
2 value corresponding to said each entry of said plurality of entries is calculated
3 based on an entropy value for said each entry and an entropy value for said
4 root entry.

1 80. The system according to claim 75, wherein said each element in
2 said lexicon database is a word.

1 81. A system for evaluating similarity among a plurality of data
2 structures, comprising:
3 a database having a plurality of entries;

4 an analyzer, coupled to said database, said analyzer configured to
5 analyze each structure of said plurality of data structure to generate at least one
6 substructure;

7 a matching unit, coupled to said analyzer and said database, said
8 matching unit configured to match said at least one substructure to at least one
9 entry of said plurality of entries to obtain at least one matching entry; and

10 an entropy calculator, coupled to said matching unit and said database,
11 configured to generate a match value using a relative entropy value
12 corresponding to said at least one matching entry.

1 82. The system according to claim 81, wherein said plurality of entries
2 are created offline using a tool having a graphical user interface and are
3 exported to said database.

1 83. The system according to claim 81, wherein said entropy calculator
2 further calculates said relative entropy value corresponding to each entry of
3 said plurality of entries.

1 84. The system according to claim 83, wherein said database stores
2 said each entry together with said corresponding relative entropy value in a
3 compressed format.

1 85. The system according to claim 81, wherein said matching unit
2 further retrieves at least one code from said at least one substructure and
3 matches said at least one code to said at least one entry to obtain said at least
4 one matching entry.

1 86. The system according to claim 81, wherein each structure of said
2 plurality of data structures is a representation of a linguistic expression.

1 87. The system according to claim 81, wherein said database is a
2 thesaurus hierarchy including a root entry, said plurality of entries depending
3 from said root entry.

1 88. The system according to claim 87, wherein said relative entropy
2 value corresponding to said each entry of said plurality of entries is calculated
3 based on an entropy value of said each entry and an entropy value of said root
4 entry.

ABSTRACT

A system and method for evaluating similarity among multiple data structures are provided. In one embodiment, a number of entries are created in a database and a relative entropy value corresponding to each entry is

5 calculated. Each data structure is analyzed and at least one substructure is generated for each data structure as a result of the analysis. This at least one substructure pertaining to each data structure is matched to the entries in the database to obtain matching entries. Finally, a match value is generated using the relative entropy values corresponding to the matching entries.

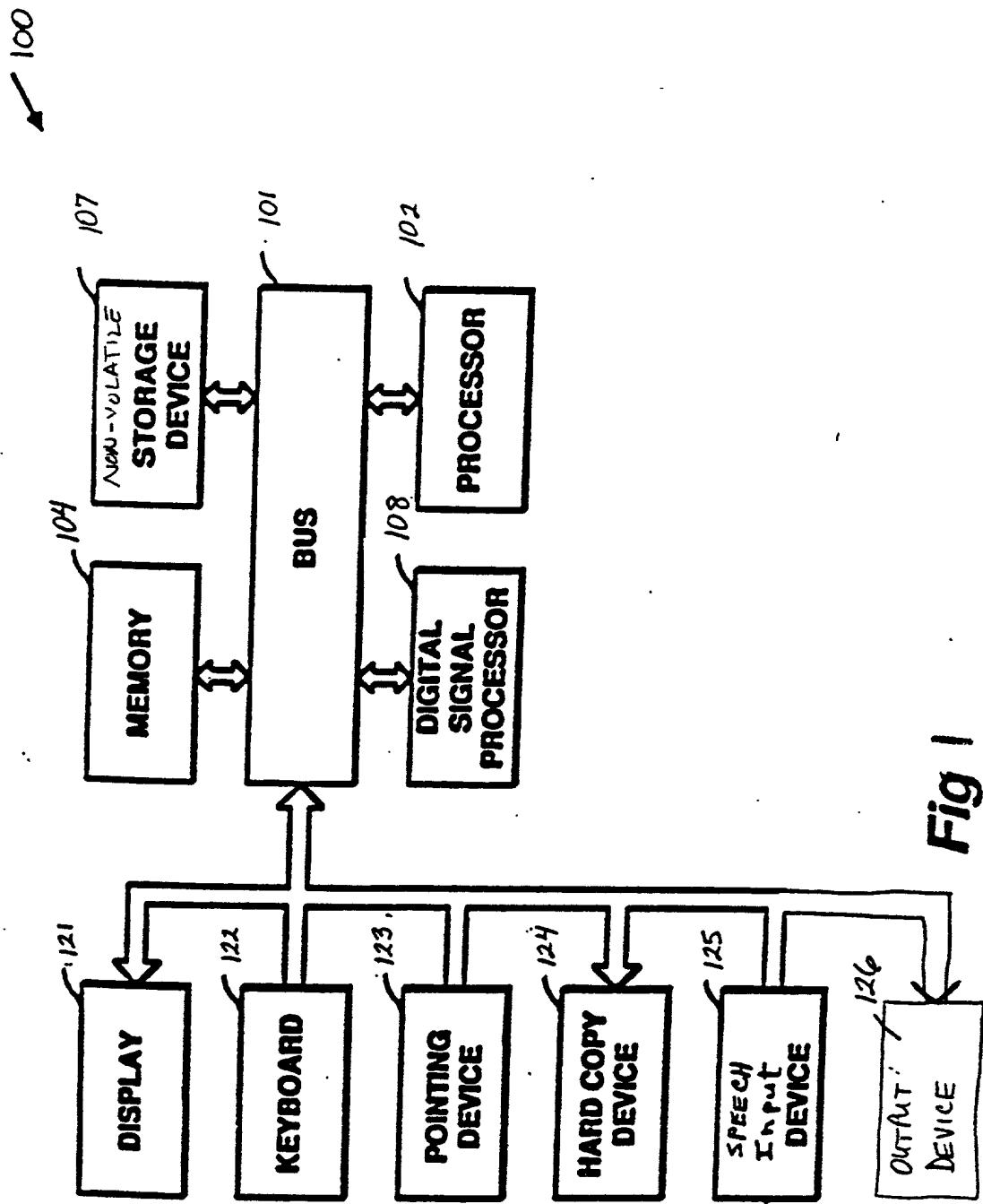


Fig 1

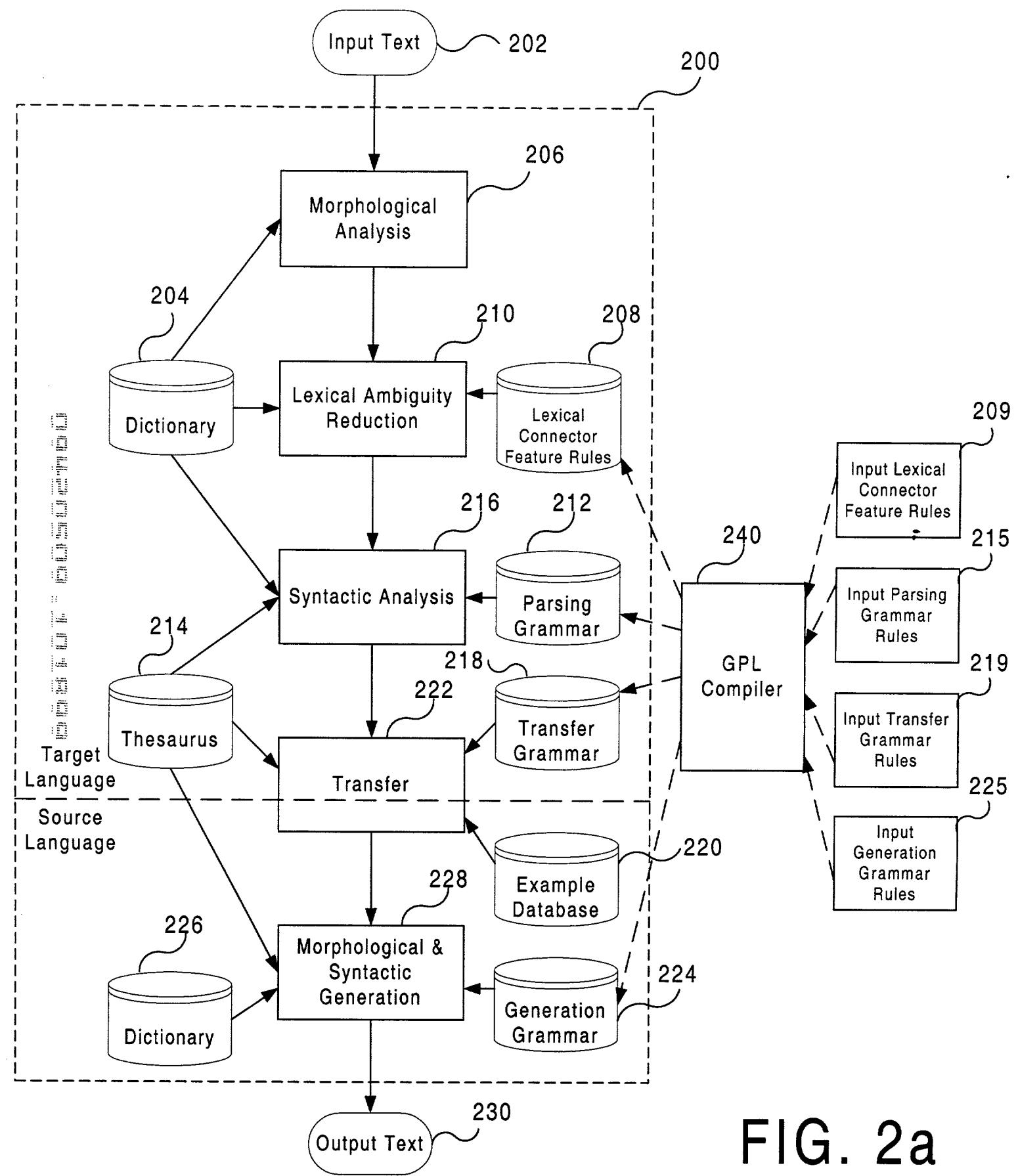


FIG. 2a

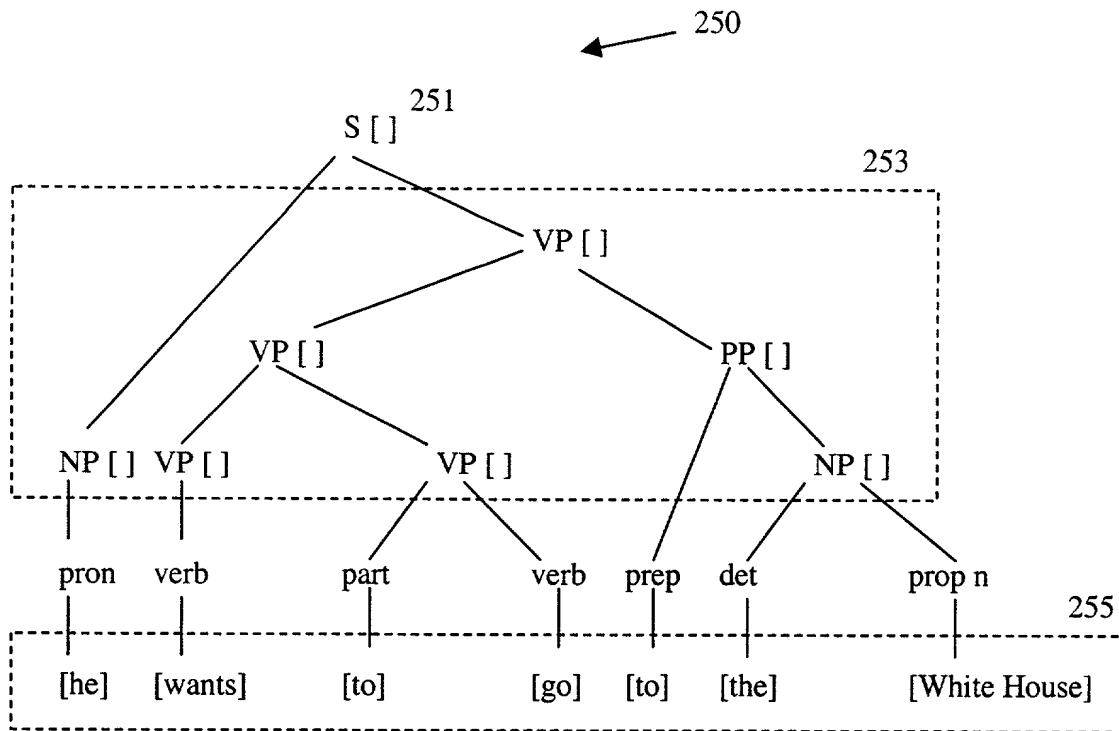


FIG. 2B

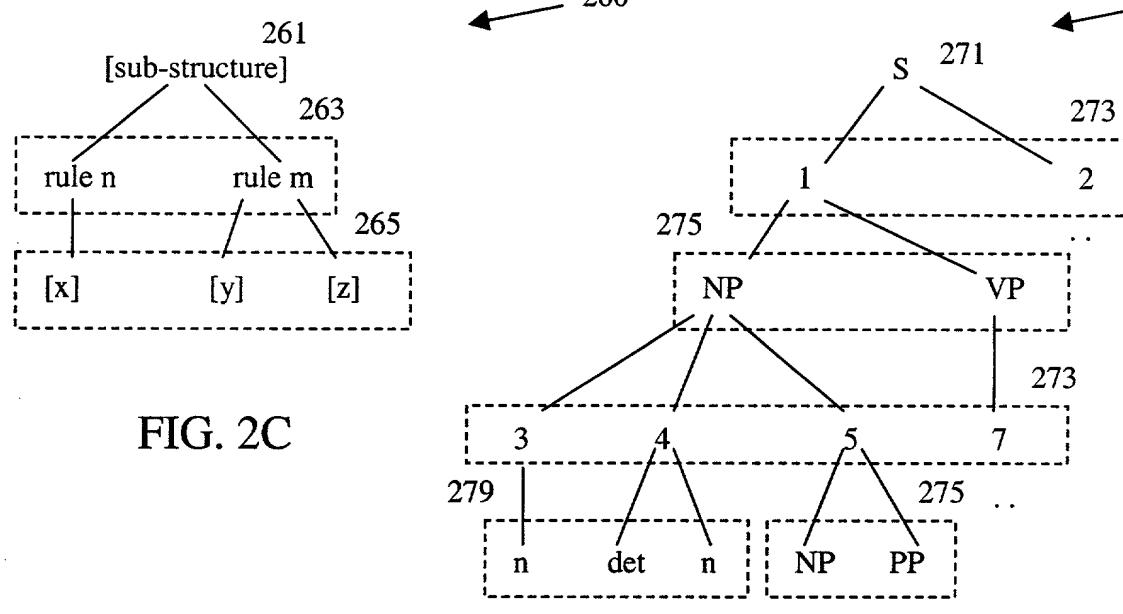


FIG. 2C

FIG. 2D

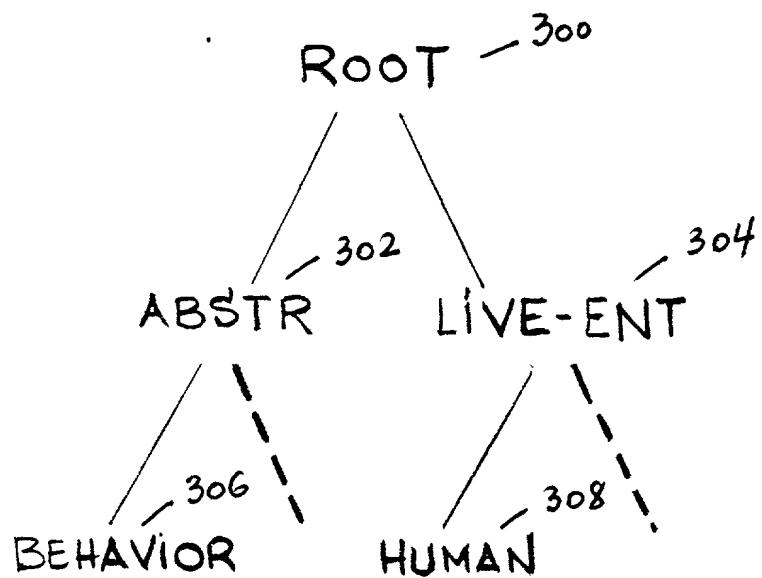


Fig 3A

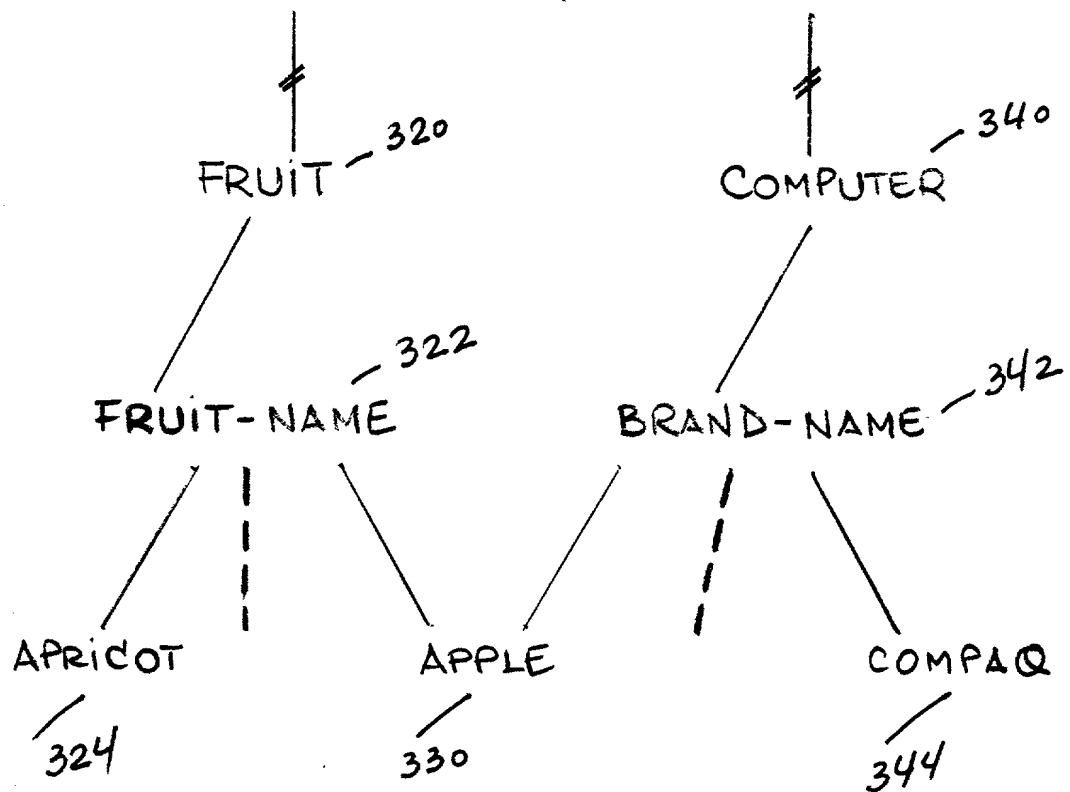


Fig. 3B

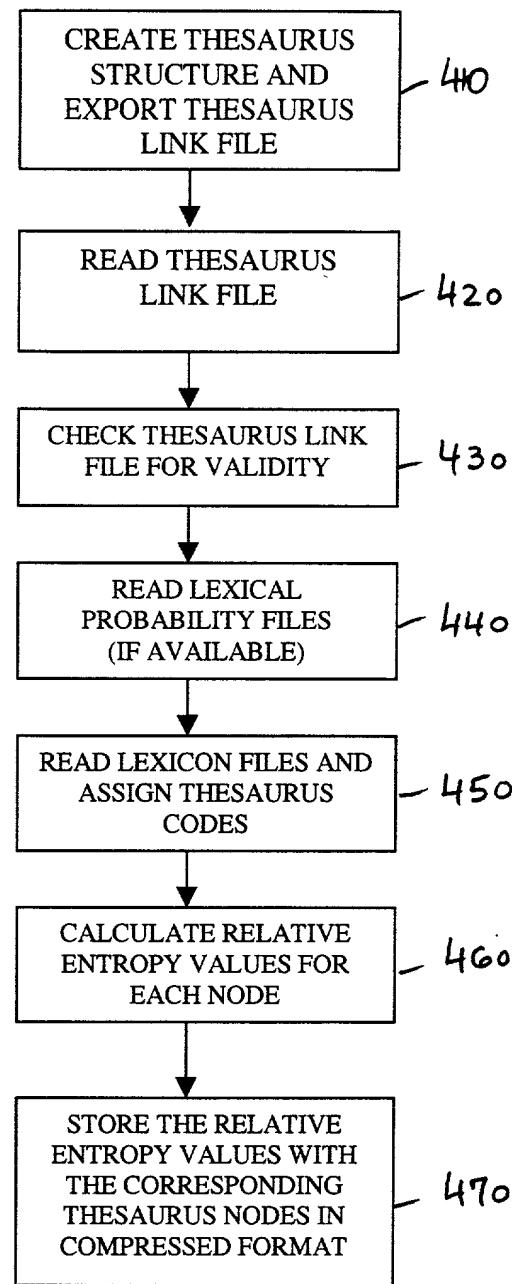


Fig. 4

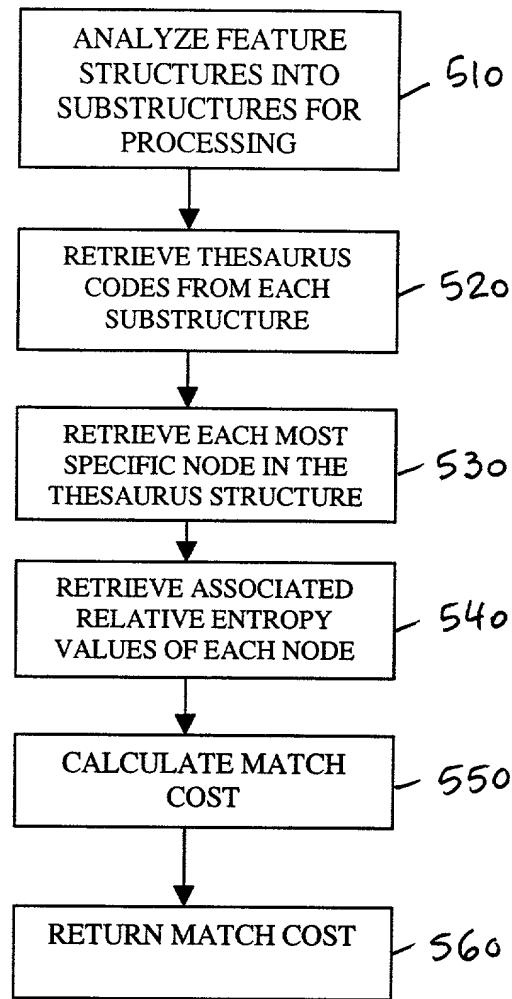


Fig. 5

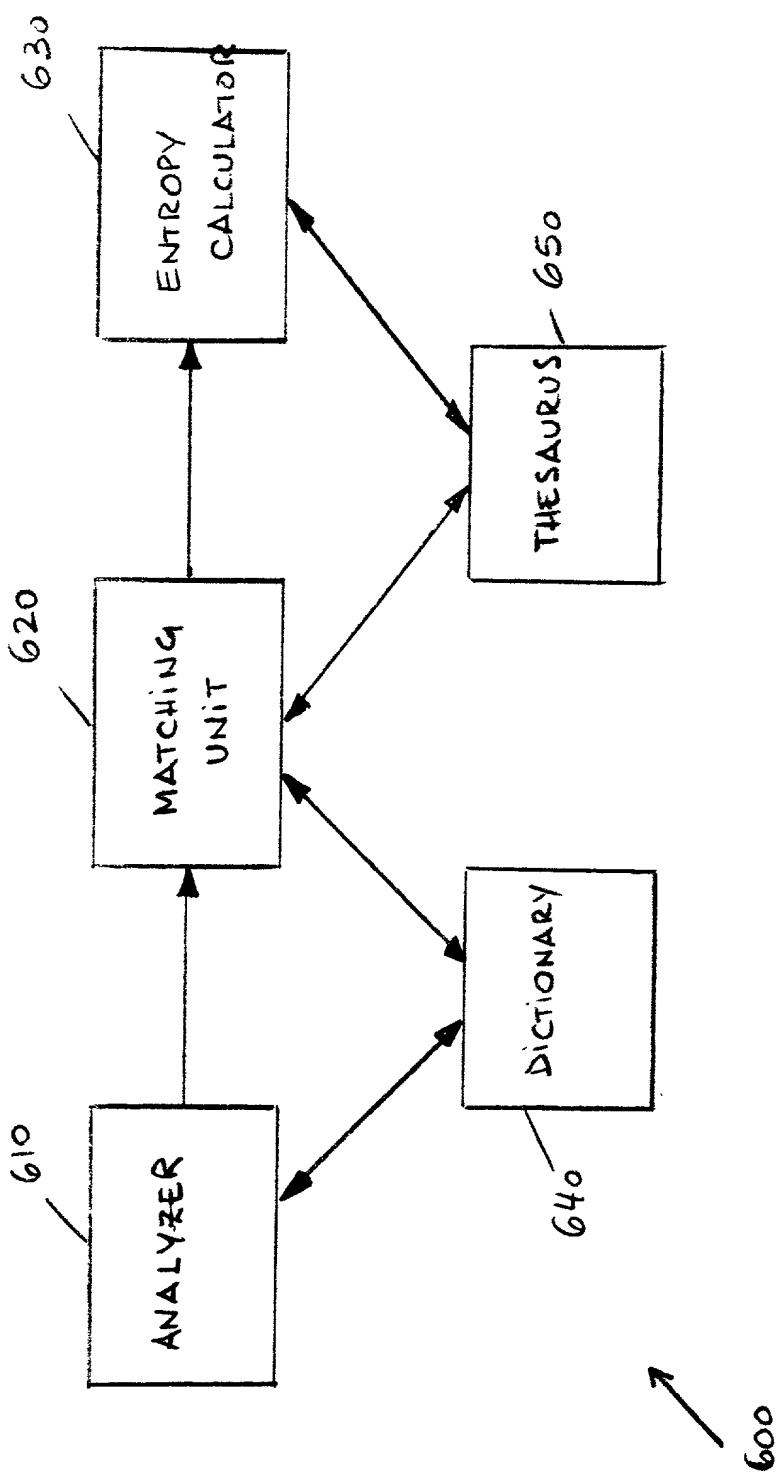


Fig. 6

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first, and sole inventor (if only one name is listed below) or an original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

SYSTEM AND METHOD TO MATCH LINGUISTIC STRUCTURES USING THESAURUS INFORMATION

the specification of which

XXX is attached hereto.
— was filed on _____ as
United States Application Number _____
or PCT International Application Number _____
and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claim(s), as amended by any amendment referred to above. I do not know and do not believe that the claimed invention was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months (for a utility patent application) or six months (for a design patent application) prior to this application.

I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d), of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

<u>Prior Foreign Application(s)</u>			<u>Priority Claimed</u>
<u>(Number)</u>	<u>(Country)</u>	<u>(Day/Month/Year Filed)</u>	<u>Yes</u> <u>No</u>

I hereby claim the benefit under title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below:

<u>(Application Number)</u>	<u>Filing Date</u>

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

<u>(Application Number)</u>	<u>Filing Date</u>	<u>(Status -- patented, pending, abandoned)</u>

I hereby appoint the persons listed on Appendix A hereto (which is incorporated by reference and a part of this document) as my respective patent attorneys and patent agents, with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

Send correspondence to Maria McCormack Sobrino, BLAKELY, SOKOLOFF, TAYLOR & (Name of Attorney or Agent)
 ZAFMAN LLP, 12400 Wilshire Boulevard 7th Floor, Los Angeles, California 90025 and direct telephone calls to Maria McCormack Sobrino, (408) 720-8598.
 (Name of Attorney or Agent)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor Alexander Franz

Inventor's Signature Alexander Franz Date 10/15/99

Residence Palo Alto, CA Citizenship Germany
(City, State) (Country)

Post Office Address 2895 Kipling Street
Palo Alto, CA 94306

Full Name of Second/Joint Inventor Keiko Horiguchi

Inventor's Signature Keiko Horiguchi Date 10/15/99

Residence Palo Alto, CA Citizenship Japan
(City, State) (Country)

Post Office Address 2895 Kipling Street
Palo Alto, CA 94306

Full Name of Third/Joint Inventor Kazami Uchida

Inventor's Signature Kazami Uchida Date 10-15-1999

Residence Mountain View, CA Citizenship Japan
(City, State) (Country)

Post Office Address 505 Central Avenue, Apt. #509G 730 East Evelyn Ave. #632
Mountain View, CA 94041 Sunnyvale, CA 94086

Full Name of Fourth/Joint Inventor

Inventor's Signature _____ Date _____

Residence _____ Citizenship _____
(City, State) (Country)

Post Office Address _____

APPENDIX A

William E. Alford, Reg. No. 37,764; Farzad E. Amini, Reg. No. P42,261; Aloysius T. C. AuYeung, Reg. No. 35,432; William Thomas Babbitt, Reg. No. 39,591; Carol F. Barry, Reg. No. 41,600; Jordan Michael Becker, Reg. No. 39,602; Bradley J. Bereznak, Reg. No. 33,474; Michael A. Bernadicou, Reg. No. 35,934; Roger W. Blakely, Jr., Reg. No. 25,831; Gregory D. Caldwell, Reg. No. 39,926; Ronald C. Card, Reg. No. 44,587; Thomas M. Coester, Reg. No. 39,637; Stephen M. De Clerk, under 37 C.F.R. § 10.9(b); Michael Anthony DeSanctis, Reg. No. 39,957; Daniel M. De Vos, Reg. No. 37,813; Robert Andrew Diehl, Reg. No. 40,992; Matthew C. Fagan, Reg. No. 37,542; Tarek N. Fahmi, Reg. No. 41,402; James Y. Go, Reg. No. 40,621; James A. Henry, Reg. No. 41,064; Willmore F. Holbrow III, Reg. No. P41,845; Sheryl Sue Holloway, Reg. No. 37,850; George W Hoover II, Reg. No. 32,992; Eric S. Hyman, Reg. No. 30,139; Dag H. Johansen, Reg. No. 36,172; William W. Kidd, Reg. No. 31,772; Erica W. Kuo, Reg. No. 42,775; Michael J. Mallie, Reg. No. 36,591; Andre L. Marais, under 37 C.F.R. § 10.9(b); Paul A. Mendonsa, Reg. No. 42,879; Darren J. Milliken, Reg. 42,004; Lisa A. Norris, Reg. No. P44,976; Chun M. Ng, Reg. No. 36,878; Thien T. Nguyen, Reg. No. 43,835; Thinh V. Nguyen, Reg. No. 42,034; Dennis A. Nicholls, Reg. No. 42,036; Kimberley G. Nobles, Reg. No. 38,255; Daniel E. Ovanezian, Reg. No. 41,236; Babak Redjaian, Reg. No. 42,096; William F. Ryann, Reg. 44,313; James H. Salter, Reg. No. 35,668; William W. Schaal, Reg. No. 39,018; James C. Scheller, Reg. No. 31,195; Jeffrey Sam Smith, Reg. No. 39,377; Maria McCormack Sobrino, Reg. No. 31,639; Stanley W. Sokoloff, Reg. No. 25,128; Judith A. Szepesi, Reg. No. 39,393; Vincent P. Tassinari, Reg. No. 42,179; Edwin H. Taylor, Reg. No. 25,129; John F. Travis, Reg. No. 43,203; George G. C. Tseng, Reg. No. 41,355; Joseph A. Twarowski, Reg. No. 42,191; Lester J. Vincent, Reg. No. 31,460; Glenn E. Von Tersch, Reg. No. 41,364; John Patrick Ward, Reg. No. 40,216; Charles T. J. Weigell, Reg. No. 43,398; Kirk D. Williams, Reg. No. 42,229; James M. Wu, Reg. No. P45,241; Steven D. Yates, Reg. No. 42,242; Ben J. Yorks, Reg. No. 33,609; and Norman Zafman, Reg. No. 26,250; my patent attorneys, and Andrew C. Chen, Reg. No. 43,544; Justin M. Dillon, Reg. No. 42,486; Paramita Ghosh, Reg. No. 42,806; and Sang Hui Kim, Reg. No. 40,450; my patent agents, of BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, with offices located at 12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025, telephone (310) 207-3800, and James R. Thein, Reg. No. 31,710, my patent attorney.

APPENDIX B

Title 37, Code of Federal Regulations, Section 1.56 Duty to Disclose Information Material to Patentability

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclosure information exists with respect to each pending claim until the claim is cancelled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is cancelled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclosure all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) Prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) The closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made or record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or

(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application;

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.